LA-UR-01-4334

Title: Extreme-Scale Architecture Simulation

Author(s): Francis Alexander, Kathryn Berkbigler,
Graham Booker, Brian Bush,
Thomas Caudell, Kei Davis,
Tim Eyring, Adolfy Hoisie,
Donner Holten, Steve Smith,
Kenneth Summers

Submitted to: SC2001 Research Poster Session,
Denver, Colorado
10-16 November 2001

Los Alamos
NATIONAL LABORATORY

Form 836 (8/00)

à la carte : an approach to extreme-scale simulation of novel architectures

F. Alexander,[1] K. Berkbigler,[1] G. Booker,[1] B. Bush,[1] T. Caudell,[2] K. Davis,[1] A. Hoisie,[1] D. Holten,[2] S. Smith,[1] K. Summers,[2] C. Zhou[2]

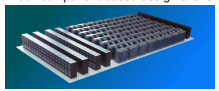
[1]Los Alamos National Laboratory and [2]University of New Mexico

http://www.c3.lanl.gov/~parsim/

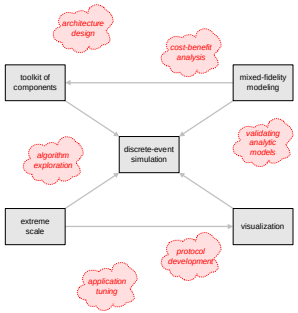
Sponsored by ASCI (Accelerated Strategic Computing Initiative) DisCom2 (Distance and Distributed Computing and Communication)

ABSTRACT

Better hardware design and lower development costs involve performance evaluation, analysis, and modeling of parallel applications and architectures, and in particular predictive capability. We outline an approach to simulating computing architectures applicable to extreme-scale systems (thousands of processors) and to advanced, novel architectural configurations. Our component-based design allows for the seamless assembly of architectures from representations of workload, processor, network interface, switches, etc., with disparate and variable resolutions into an integrated simulation model. Our initial prototype, comprising low-fidelity models of workload and network, easily scales to many thousands of computational nodes in a fat-tree network.



GOALS

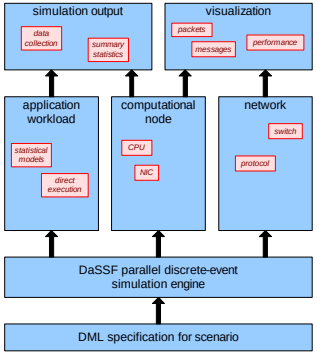


Problem: The magnitude of the scientific computations targeted by the ASCI project requires as-yet unavailable computational power. Current approaches to building larger supercomputers—connecting commercially-available SMPs with a network—may be reaching practical limits.

In response, the DOE Advanced Architecture Initiative seeks to research alternative high-performance computing architectures.

The *à la carte* project aims to develop a simulation-based analysis tool for evaluating massively-parallel computing platforms including current and future ASCI-scale systems. Such a tool will provide a means to analyze and optimize the current systems and applications as well as influence the design and development of next-generation high-performance computers.

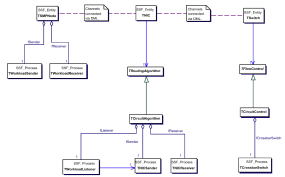
APPROACH



Our component-based design (*see figure at left*) allows for the seamless assembly of architectures from representations of workloads, processors, network interfaces, switches, etc., with disparate resolutions, into an integrated simulation model. One can mix and match components of different fidelities to construct a model with the appropriate level of detail for a particular study. We are focusing on the development of a simulation capability that scales to tens of thousands of processors and that can execute on a wide variety of computing platforms.

Simulating systems of the size and complexity we envision requires efficient parallel simulation. We use a portable, conservative synchronization engine (DaSSF), developed by Dartmouth College, for the handling of discrete events. DaSSF manages the synchronization, scheduling, and delivery of events in the simulation; it has a lean C++ API and supports both shared-memory and distributed-memory parallelism. We use Domain Modeling Language (DML) to specify the architecture and workload to be simulated. DML allows one to easily construct libraries of reusable component specifications.

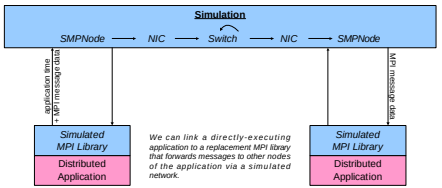
The initial prototype (*see figure at right*), comprising low-fidelity models of workload and network, models at least 4096 computational nodes in a fat-tree network. This prototype supports studies of simulation performance and scaling rather than the properties of the simulated systems themselves.



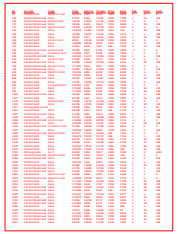
Ongoing work in our iterative development approach aims to improve the fidelity of the representations and protocols. Future work will emphasize validation, the representation of I/O and storage, and wide-area networking.

Applications and computational workloads may be represented at a variety of fidelities. Each approach below addresses tradeoffs between the accuracy of the model and the computing resources required for the model.

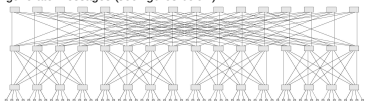
- Simple random processes can load the hardware with message traffic having specified statistical properties. These match the distribution of messages in a real application and can include temporal and spatial correlations between messages. They ignore some of the data dependencies, however.
- Direct-execution techniques (*see figure at right*) allow one to run programs nearly exactly on real processors coupled to a simulated network. These are faithful to the actual timing of an application on a processor, but may be very computationally intensive or slow.
- From time series of fine-grained simulations we will use learning algorithms to construct reduced models of the full system dynamics. This involves regression techniques like neural networks or dimension reduction methods such as the Karhunen-Loeve expansion.

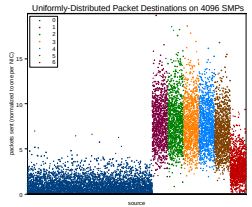


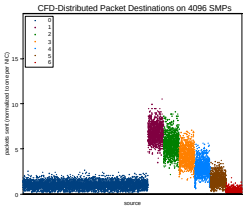
ANALYSIS



Our simulation provides complete detail concerning the history of messages and the propagation of packets through the simulated network (*see figure at left*). Data summaries supply information on queue sizes, throughputs, port usage, timeouts, path lengths, and communication patterns. Output is configurable in terms of when data is collected and what is collected. We are currently running simulations of a 4096-node fat-tree containing 6144 switches using a basic circuit-switched protocol and simple random process to generate messages (*see figures below*).







Scientific visualization techniques are used to analyze output as well as debug the simulations. We can study network behavior/performance and the communication patterns and network usage of applications. Our visualization approaches include direct representations of the architecture as well as innovative abstractions of the architecture and dynamics of the system (*see figures below*).